Persistent Aerial Tracking using UAVs*

Matthias Mueller[1], Gopal Sharma[2], Neil Smith[1], Bernard Ghanem[1]

***Abstract*— In this paper we propose a persistent, robust and autonomous object tracking paradigm called Persistent Aerial Tracking (PAT). A computer vision and control strategy for PAT is applied to a diverse set of moving objects (e.g. humans, animals, cars, boats, etc.) integrating multiple UAVs with a stabilized RGB camera. A novel strategy is employed to successfully track objects over a long period, by multiple UAVs 'handing over the camera' from one UAV to another. Due to the lack of an extensive aerial video dataset and benchmark for low altitude UAV target tracking, this paper provides the first evaluation of target trackers on 64 new fully annotated HD video sequences all captured from a professional grade UAV. Based on our evaluations, we select the leading tracker and improve upon it by optimizing for both speed and performance, integrate the complete system on an off-the-shelf UAV and obtain promising results showing the robustness of our solution in real-world scenarios.**

I. INTRODUCTION

The ability to capture stabilized high resolution video from low-cost and accessible unmanned aerial vehicles (UAVs) has the potential to significantly redefine future objectives in the development of state-of-the-art object tracking methods. In this paper, we propose a persistent, robust and autonomous object tracking paradigm called Persistent Aerial Tracking (PAT) (see Fig. 1). Persistent aerial tracking can serve many purposes, not only related to surveillance but also search and rescue, wild-life monitoring, crowd monitoring/management, and extreme sports. Within the hobbyist community and general public the concept has become popularized as autonomous aerial selfies even though most of these approaches rely on the tracked individual carrying a GPS leash. Persistent aerial tracking (PAT) using UAVs is a very promising application, since the camera can actively follow the target based on visual feedback it generates, thus, *actively* changing its orientation and position to optimize for tracking performance (e.g. persistent tracking accuracy in the presence of occlusion or fast motion across large and diverse areas). This poses the defining difference with static target tracking systems, which passively analyze a dynamic scene to produce analytics for other systems. It enables ad-hoc and low-cost surveillance systems that can be quickly deployed, especially in locales where a surveillance infrastructure is not already established or feasible (e.g. in remote locations, rugged terrain, and large water bodies).

Although a few large annotated video datasets have been recently published with visual tracker benchmarks, none of these datasets are captured from a mobile aerial platform. It is not clear from these works which trackers would perform most efficiently where many tracking challenges are amplified, including rapid camera motion, significant changes in scale and camera orientation, fast moving objects that until now could not be followed by static cameras, occlusion (e.g. natural and urban features), and the requirement to quickly translate tracked information to appropriate UAV control. In order to provide a benchmark on current target trackers and open new avenues for the research community to develop better algorithms for persistent aerial tracking applications, this paper provides the first evaluation of target trackers on 64 new fully annotated HD video sequences all captured from a professional grade UAV. We anticipate this dataset will provide a baseline that can be used long into the future as UAV technology advances and target trackers improve.

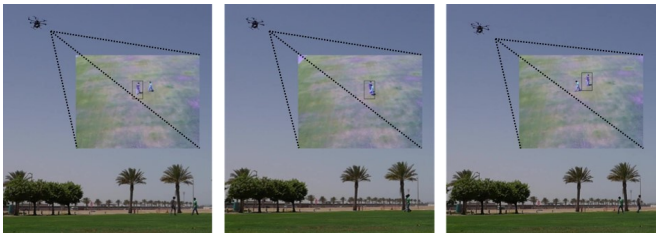

Fig. 1: UAV visually tracking a moving human despite occlusion

A current drawback of UAV use for *persistent* aerial tracking is their limited flight time (ca. 10-25 minutes) especially with multi-rotor copters. We propose a novel strategy to address the problem of persistent autonomous aerial tracking, by introducing the concept of target tracking handover among a network of coordinated UAVs. In this scenario, 'camera handover' is a process similar to what is needed in traditional static tracking systems, since it involves transferring the tracked targets appearance model from one fixed camera to another. However, in the case of PAT, camera handover also requires the exchange of flight data of the active UAV (e.g. GPS coordinates, altitude, heading, etc.) to one or more UAVs. When a UAV reaches its first low battery threshold it can communicate to another UAV to move into location and resume tracking. Moreover, since communication range and transmission speed can prove to be bottlenecks when relying on a ground station for online tracking, our approach allows all computations to be conducted onboard each UAV. The ground stations primary role is to assist human operators in initially selecting what they want the UAV to track, sending the UAV to new vantage points for tracking and to receive its monitored feedback.

Our computer vision and control strategy for PAT can be applied to a diverse set of moving objects (e.g. humans,

*This work was supported by KAUST, Saudi Arabia

[1]Matthias Mueller, Neil Smith and Bernard Ghanem are with the Department of Electrical Engineering, KAUST bernard.ghanem.2@kaust.edu.sa

[2]Gopal Sharma with Dept. of Electrical Engineering, IIT Roorkee, India

animals, cars, boats, etc.) all using multiple UAVs with a stabilized RGB camera. The proposed method is validated by a series of experiments, where humans and cars are tracked in outdoor cluttered environments with a variety of appearance and scale changes. Based on extensive evaluations and tracker comparisons on our large benchmark, we identify a suitable tracker, improve it for aerial tracking, and integrate it into a completely automated UAV system. This system obtains very encouraging results showing the robustness of our solution in real-world aerial scenarios.

The contributions of our work include:

1) A fully annotated high-resolution dataset of 64 aerial video sequences with more than 23,000 frames. This is the largest aerial tracking dataset. It is as large as the most recent generic, object tracking datasets [1]. On this dataset, we benchmark 8 popular trackers using multiple evaluation metrics [2]. Based on the benchmark results, we select the leading tracker and improve upon it by optimizing for both speed and performance.
2) An integrated UAV system that performs autonomous onboard aerial tracking using the best performing object tracking algorithm. The tracking system autonomously controls the UAV and relays tracked results to a ground station. The system is fully modular with the ability to modify the tracking technique and is mountable on commercially available UAVs.
3) A novel strategy for camera handover to track objects persistently and a re-initialization module in case the target is lost.

Related Work

Early object tracking methods for UAVs primarily rely upon two approaches. The first approach applies Canny Edge Detectors and Harris Corner Detectors to isolate distinct feature points then accumulative frame differencing methods and background subtraction for blob tracking of all moving targets within the UAVs FOV [3][4][5][6][7][8][9]. Tensor voting and motion pattern segmentation is used by [9] to address parallax, noise in background modeling and long term occlusions. More recent work by [10][11] focuses on tracking for sense and avoid maneuvers utilizing a combination of feature point tracking and morphological close-minus-open filters. They first divide each frame between terrain and sky, then apply the morphological operations to the sky and feature detection to both. The morphological detector enhances performance with larger detection distances for the sky region. A major drawback of the tracker is that it falsely identifies lens flare as aircraft. This is overcome in later work by the authors [11]. In [12], SIFT is used to detect salient Keypoints that are then tracked across frames. Although SIFT features are invariant to scale, minor light variation and affine changes, the test evaluations only track static targets and poorly handle targets with homogenous and/or recurrent shapes. Drawbacks of these feature-based approaches are discussed in several papers; they poorly handle any form of minor occlusion, light change, or homogenous features indistinguishable from the background.

The second approach utilizes color information to detect and track moving targets [8][13][14][15][16]. These works [8][13][14][15] develop robust vision-based control mechanisms to control the flight motion of the UAV, known as visual servoing. Continuously Adaptive Mean Shift is predominately used for tracking. In order to improve tracking with minor occlusions and fast movement, [15] apply a multi-kernel representation along with particle filtering to better predict the location of the target. In all of these studies, the object tracked is a large uniform color (e.g. red balloon, red car). This approach is extremely limited to scenarios where distinct colors of the target are clearly distinguishable from the background and there are no abrupt scale change, major occlusion, complex motion or perspective change.

Other work uses thermal and IR cameras as an alternative to RGB cameras [17][18][19]. Humans and cars' thermal signature can be easily distinguished from the background using approaches such as Mean-shift. [18] and [19] train HAAR classifiers to isolate only the human body signatures, speed up computation, and account for varying illumination, base color and scale. [19] also introduces a new dataset of thermal imagery including 4381 manually annotated images captured from an elevated platform. They evaluate their dataset with multiple trackers showing a marked improvement with the use of their HAAR classifier and included particle filter.

Target tracking methods are also being applied to UAV captured video datasets of wide aerial video [20][21][22]. The objectives of tracking in wide aerial video are different from this paper but show promising methods that will help in initialization and more intelligent target understanding. [20] notes that a major disadvantage of the target detection method applied in these papers is the reliance on frame differencing that cannot maintain persistent tracking once the objects stop moving. As a solution, they employ an appearance-based regressor. [21] provide the first fully annotated UAV captured dataset for evaluating wide aerial video. Two top-down scenes are captured of a picnic area where diverse group events and human roles occur. By applying Markov Chain Monte Carlo they are able to detect multiple moving targets and recognize temporal extents of events and human roles.

Most closely related to our approach is work by [23] and [24]. In [23], the object tracker TLD [25] is used to track objects from a UAV and obtains good short-term results. However, as their experiments show, TLD incorporates the background into its learning over time, leading to target loss after only one minute of tracking. The experiments only track a person at an altitude of ca. 1.5 m, not taking into consideration common tracking problems such as large perspective change and scale variation. The second paper [24] presents a new tracker outperforming MIL [26] and TLD [25] for visual aircraft model-free tracking. The tracker is able to track another aircraft/intruder in the sky with illumination changes (strong sunlight) and background clutter (clouds) from a fixed-wing UAV. Note, that the experiments are very application specific with object (aircraft/intruder) and uncluttered background (sky) always being the same.

Therefore, while these contributions are related to our work, they consider specific applications with a different intention as our proposal and rely on trackers that are significantly out performed by more recent work (see below).

A review of related work suggests that there is still a limited availability of annotated datasets specific to UAVs in which trackers can be rigorously evaluated for precision and robustness. Existing annotated video datasets include very few aerial sequences [1], [2]. Surveillance datasets such as PETS or CAVIAR focus on static surveillance and are outdated [27], [28], [29]. One of our paper's major contributions is the capture and annotation of a diverse high-resolution data set for tracking from UAVs and a thorough evaluation of state-of-the art trackers for persistent aerial tracking. This study evaluates classical trackers such as OAB [30] and IVT [31] as a baseline and the best-performing recent trackers according to [2]: Struck [32], CSK[33], ASLA[34], CXT[35] and TLD[25]. In the selection process, we reject very slow trackers that perform poorly in previous benchmarks such as SCM [36] and avoid trackers with similar methods. In addition, we include one of the latest trackers based on the MEEM framework [37]. This new benchmark will guide research to develop new aerial-based target trackers and allow researchers to participate that may not have access to hardware or permitted safe-fly zones.

II. Our Proposed Method

A. Data Set

The dataset was captured from an off-the-shelf professional-grade UAV (DJI S1000) with a fully stabilized and controllable gimbal system (DJI Zenmuse Z15). UAV video sequences were recorded using a Panasonic GH4 with Olympus M.Zuiko 12mm f2.0 lens. The UAV follows different objects at altitudes varying within 5-25 meters.

Our dataset contains a total of 64 sequences generated from 16 continuous shots. The collection contains diverse scenes of urban landscape: roads, buildings, fields, beaches and a harbor. The sequences are 720p at 10fps, completely annotated with upright bounding boxes. By design, these video sequences contain common visual tracking challenges including long-term full and partial occlusion, scale variation, illumination variation, perspective change, background clutter, camera motion, object motion, etc. Table I shows an overview of all attributes used to annotate each sequence and Figure 2 shows the distribution of these attributes over the whole dataset. Table II shows the highest available frame rate and resolution for each sequence when originally recorded.

B. Benchmark

For fair evaluation, all trackers are run with standard parameters. To identify optimal tracking methods for UAVs, the following trackers are compared: IVT [31], CXT[35], TLD[25], MEEM [37], Struck [32], OAB [30], CSK[33], and ASLA[34]. In addition, we compare $\text{Struck}_{\text{UAV}}$, our proposed improvement on Struck for UAV applications. All trackers are tested on the same server-grade workstation (Intel Xenon X5675, 3.07GHz, 48GB).

TABLE I: Attributes used to annotate each sequence.

Attr	Description
IV	Illumination Variation: illumination of target changes significantly.
SV	Scale Variation: the ratio of initial bounding box and at least 10% of subsequent bounding boxes is outside the range [0.5, 2].
POC	Partial Occlusion: the target is partially occluded or part of the object leaves the view.
FOC	Full Occlusion: the target is fully occluded or the complete object leaves the view.
OV	Out-of-View: some portion of the target leaves the view.
FOM	Fast Object Motion: fast motion of the target.
FCM	Fast Camera Motion: fast motion of the camera.
BC	Background Clutter: background has similar appearance as the target.
SOB	Similar Object: there are objects of similar shape or same type near the target.
PC	Perspective Change: perspective affects target appearance.
LR	Low Resolution: the ground-truth bounding box has less than 1600 pixels in at least 50% of all frames.

TABLE II: Pixel resolution and frame rate for each recorded sequence.

	boat	car1-3	group1	group2-3	person1-6	person7-9
Res	1080p	1080p	720p	1080p	720p	1080p
FPS	30	96	30	96	30	96

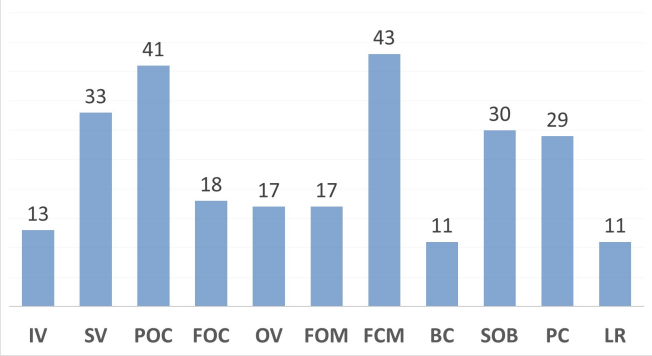

Fig. 2: Attribute distribution across dataset.

Following the evaluation strategy of a recent generic tracking benchmark [2], all trackers are compared using two measures: precision rate and success rate. The precision plot shows the percentage of tracker bounding boxes within the given threshold distance in pixels of the ground truth. To rank the trackers, we use a threshold of 20 pixels [2]. The success plot shows the intersection over the union of the tracker and ground truth bounding boxes. Given an overlap threshold, this metric shows the percentage of tracker bounding boxes within the given overlap threshold. The trackers are ranked with an overlap ratio of 0.5. Besides one-pass evaluation (OPE), we perform a spatial robustness evaluation [2]. For spatial robustness evaluation (SRE), the initial bounding box is spatially shifted by 4 center shifts and 4 corner shifts and scaled by 80, 90, 110 and 120 percent [2].

C. Improvements on Struck

Our evaluation shows that the Struck tracker [32] performs best in terms of tracking performance to speed ratio. Struck reaches up to 20fps and is able to handle many tracking challenges (e.g. partial occlusion). Therefore, to overcome the

limitations in computational power of the on board computer and to further improve performance, we optimize Struck for UAV applications. Struck (Structured Output Tracking with Kernels) [32] is developed by Sam Hare *et al.* and is open source. Most trackers approach the tracking problem as a classification task and use an online classifier to build an object appearance model. Estimated object positions are converted into labeled training instances. Struck bridges the gap between the objectives of the classifier (label prediction) and tracker (accurate prediction of object location) by learning a prediction function that directly outputs the transformation of the bounding box from one frame to the next. The prediction function is learned online to adapt the object appearance within a kernelized structured output SVM framework. For real-time applications, a budgeting mechanism is incorporated to bind the growth of support vectors during tracking. Struck has notable performance even when the target moves fast [2] because of the dense sampling of a large search region that provides better target discrimination from the background. Struck is less sensitive to scale variation and performs well even when the initial bounding box is not strictly tight.

a) Speed and scaling: We improve the Struck tracker's speed and scale adaptation to better prepare it for aerial video recordings, where the target and/or camera move quickly and the target scale can change drastically due to the moving viewpoint change of the UAV. We call our improved version of the tracker Struck$_{\text{UAV}}$. Many Struck parameters that determine the search radius and how the classifier updates are already suitable for the UAV. However, for faster tracking, we reduce the number of samples we search over to find the optimal. Also, we refrain from performing exhaustive search in a local neighborhood to update the classifier, but instead use a generous 3 pixel step size to reduce the number of evaluations the tracker performs. These minor adjustments increase the speed of tracking significantly without much loss in accuracy.

In order to improve accuracy in Struck$_{\text{UAV}}$, we make it scale adaptive. At each frame, we generate a grid of different size bounding boxes which are added to the array of bounding boxes scored by Struck, allowing for bounding box resizing in case of scale variation. This improves updates, since it avoids the classifier being updated with only small parts of the object and too much background. Even though Struck has some robustness to scale variation, we found that a more dynamic model to handle scale changes improves tracking performance even further. Figure 3 highlights the improvement in bounding box scaling of Struck$_{\text{UAV}}$ compared to TLD, Struck, and MEEM. Struck$_{\text{UAV}}$'s bounding box tightly bounds the tracked object not only allowing a higher accuracy in centering the bounding box over the target, but also handles change in target shape and appearance better. This is clearly seen in the boat example as the UAV pans from rear view to side view. It is also important to note that TLD, the other tracker implementing bounding box scaling, fails in several circumstances or does not consistently resize its bounding box with change in scale or shape.

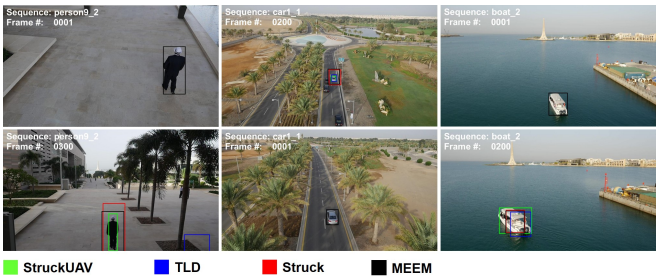

Fig. 3: Struck$_{\text{UAV}}$ and TLD with scaling ability vs. Struck and MEEM without scaling ability.

b) Target Initialization.: The original approach in [32] is to train Struck by having the tracked target hold in place within a defined bounding box for multiple frames. Since this is not feasible on a constantly moving UAV, we implement an initialization module for Struck with template matching. Tracking starts by sending a region of interest (cropped template image of the target) from the ground station to the UAV in GPS hold over the area of interest. The UAV computer performs a form of template matching called zero-mean matching, whereby the zero-mean template is used as a linear filter to compute a response map over the whole image (after making each image patch support have zero mean). The pixel location with the highest template filter response in the map is retrieved as the initial bounding box of the target. In return it is used to initialize the Struck tracker. Although other types of template matching could be used (e.g. normalized cross correlation or NCC), this method is faster and suitably accurate for our aerial tracking objectives.

c) Re-initialization.: Struck quickly learns new appearances of the object being tracked and updates the support vectors accordingly. This capability is favorable for tracking and especially in the case of UAVs where appearance, perspective and scale variation of the target are common problems. However, the tracker is not able to recover from the case of full occlusion or the target leaving the field-of-view of the camera. We implement an independent measure to provide a threshold for every new prediction. If the score from this measure drops below the threshold, we signal that the object was lost and start a global search for the object to re-initialize tracking. We incrementally build an appearance-based classifier for the target while it is being tracked, so that this classifier can be used for re-initialization in case the target is lost. While the target is being tracked, we compile a set of positive and negative training samples based on the previous results. Since training samples arrive sequentially, we learn an incremental linear SVM with HOG and color histogram features (108 features in total) [38]. The trained model provides a confidence score for the existence of the target in the bounding box predicted by Struck$_{\text{UAV}}$. If this score falls below a predefined threshold, Struck$_{\text{UAV}}$'s bounding box prediction is rejected and global search initiated.

Since exhaustive search by a sliding window is very slow, we make use of a time efficient and object-centric sampling of the entire image, namely object proposals [39]. Not only does this reduce the number of search windows to be evaluated, it does so at a high target recall rate. In

other words, the proposals generated in an image have been designed to respond to locations in the image where a generic object might be. The use of object proposals has become a standard first-step in state-of-the-art object detection systems [40]. Many object proposal methods exist in the literature, we use geodesic object proposals [39] since this method is computationally inexpensive and among the top performing proposal techniques in computer vision [41]. To reduce this method's number of proposals, we reject all proposals with a low 'objectness' score or an area that three times larger than the most recent target scale. To improve target position localization within the proposals, we sample target-sized bounding boxes within the proposal and evaluate our target SVM classifier. The bounding box inside a proposal with a classification score above a predefined threshold is selected as the new target location and the $\text{Struck}_{\text{UAV}}$ tracker is re-initialized. If this threshold is not exceeded (e.g. if the object does not re-enter the field of view), this procedure is repeated for the next frame until the object is re-identified. Fig. 4 shows an example of how we search for the target within each proposal, while Fig. 5 shows examples of re-initialization in different aerial video sequences.

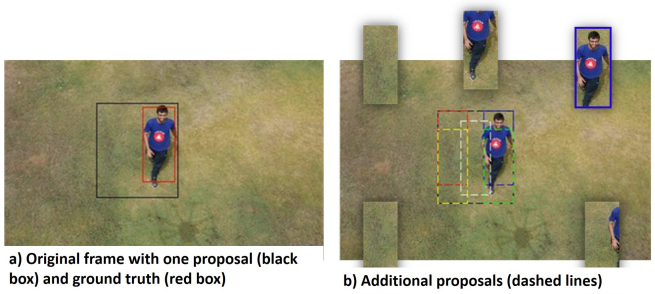

Fig. 4: (a) original frame showing one detected proposal in black and the ground truth target location in red. (b) multiple candidate bounded boxes are sampled inside the proposal and the target classifier is applied on each.

Fig. 5: Examples of the re-initialization module in action. (a) and (b) show cases where partial and full occlusion make the tracker lose the target and re-initialization is instantiated. Red bounding boxes signify that the best scoring bounding box (within proposals) has a score that falls below the threshold. Green boxes signify that the score exceeds the threshold and the $\text{Struck}_{\text{UAV}}$ tracker can be re-initialized.

D. System Integration

System Overview.

Our system consists of two UAVs: an 850mm Hexacopter with a 3-axis gimbal system and a 450mm class quadcopter with a pan/tilt gimbal. Both utilize the Pixhawk flight controller (FC) for stabilization and control of the UAV and gimbal. Onboard processing for tracking, handover and communication is handled by an ARM-based Linux computer (Odroid XU-4). Attached to the onboard computer is a USB webcam, Wifi module and FTDI to communicate with the FC. The groundstation connects by ethernet to a Wifi AP for communication with both UAVs. Figure 6 shows on overview of the complete system.

The software for the onboard computer and groundstation is written in C++ using QT5 for Linux and Windows. The communication module consists of two parts:
1) TCP Client/Server module for communication between the UAVs and the ground station. The onboard computer opens a TCP Server so that the ground station can send messages to control the UAV, while simultaneously the onboard computer sends messages to the groundstation such as current tracking status, target patches and flight data.
2) Serial communication module from the onboard computer to the Pixhawk flight controller using the Mavlink protocol.

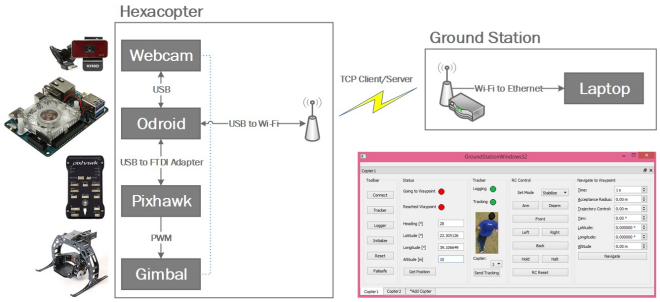

Fig. 6: System Overview

E. Camera Handover

In order to achieve persistent aerial tracking, we implement a method of camera handover. We apply a simple but robust strategy through the use of the on board GPS, compass and barometer and known camera angle of the first UAV. Once the battery of the active UAV reaches its first low voltage threshold, a handover request is transmitted to the ground station (GS) with current GPS position, altitude, heading and the tracked target's model. The GS tracks multiple UAVs simultaneously, seamlessly passing information between them. In order to initialize the tracker of the new UAV, it has to be in the proximity of the active UAV and face the desired object while keeping enough distance to avoid a collision. We calculate the horizontal distance x_{obj} between the active UAV and target using the camera angle ϕ and altitude of the UAV h_{gnd}.

Next we determine the necessary heading offset θ, given the desired distance between the active and new UAV x_{uav}.

Figure 7 illustrates the calculations. We then calculate the desired GPS coordinates for the new UAV using the haversine formula [42].

The new UAV receives the requested waypoint and navigates to the determined GPS position at a higher altitude than the current UAV to avoid any kind of interference. Once in position, it descends to the same altitude as the active UAV

Fig. 7: Camera Handover

and turns to the correct heading $\theta_{new} = \theta_{active} + \theta$. Now the new UAV uses the passed initialization module to identify the object in the current frame. In case the new UAV is unable to identify the object due to some error in altitude, heading or position, we search from -15 degrees to +15 until the object is found. Upon success, the new UAV starts tracking and signals the active UAV to return to home.

III. Results

A. Offline Evaluation

In order to identify the best tracker for integration on the UAV, we evaluate 8 top performing trackers on our annotated tracking dataset. We select these trackers based on their performance in the recent generic-object tracking benchmark [2]. Each tracker processes over 23,000 frames from 64 sequences, each with a variety of attributes as shown in Table I. When the attributes are considered collectively or individually, there is no clear top-performing tracker, since some trackers do very well for some types of tracking challenges (e.g. occlusion), but do not perform as well when faced with others (e.g. scale variations). However, there are a few trackers that are consistently among the top-performers across the different attributes. We identify them in what follows.

To compare the different trackers on UAV sequences, we use the same evaluation paradigm proposed in the generic tracking benchmark [2], namely one-pass evaluation (OPE) and spatial robustness evaluation (SRE). They test the sensitivity of each tracker to shift/scale changes in the target, which are encountered regularly in aerial video. In fact, SRE is especially important to evaluate our proposed system because target initialization and re-initialization can lead to bounding boxes that are not tight around the target, thus, including more background clutter and noise in the target appearance model. As such, SRE tests the sensitivity of the tracker to noisy initializations. Moreover, we do not make use of temporal robustness evaluation (TRE) in this paper, since user re-initialization should not occur in fully autonomous tracking scenarios.

The extensive experiments on the new dataset (refer to Fig. 8) show that more recent trackers outperform the classical ones by a significant margin. In the OPE precision plots, ASLA [34], CXT [35], and IVT [31] achieve similar low performance and CSK [33] and TLD [25] achieve mediocre performance, while Struck [32], Struck$_{\text{UAV}}$, and MEEM [37] are the top performers. The best performing tracker in OPE and SRE over the complete dataset is clearly the 2015 released MEEM tracker. Note that in our benchmark every 10 frames correspond to one second in real-time. Each tracker predicts a bounding box for each frame regardless of their actual speed. Of course, this is very different when tracking in real-time. If frames are not processed fast enough in-between, frames are lost resulting in larger displacement of the target between frames making tracking more difficult. Therefore, if the tracker is too slow, the tracking performance will degrade. Even if the tracker can cope with lower frame rates, updates for the UAV will be less frequent making it more difficult to keep the moving target within the field-of-view (FOV). The target can be easily lost and the UAV will be forced to hold in place searching to re-initialize the lost target. In our experiments, we run each tracker on the same server-grade workstation. In such a setup MEEM achieves only 8 fps, while Struck$_{\text{UAV}}$ reaches 21 fps and achieves the second best performance overall. By factoring both performance and speed together, Struck$_{\text{UAV}}$ outperforms all other trackers. Comparing Struck$_{\text{UAV}}$ to the original Struck the effect of the improvements can be clearly seen, as Struck$_{\text{UAV}}$ outperforms the original Struck by a margin while maintaining an even higher frame rate.

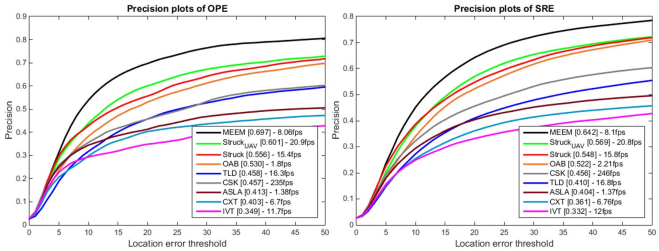

Fig. 8: Precision plot for OPE and SRE over complete data set.

A very common challenge for tracking is scale variation, which occurs in more than half of the sequences in the dataset. A tight bounding box around the tracked target is critical in persistent aerial tracking for determining follow distance and the proportional response speed required for the UAV to keep the target centered within the FOV. In the subset of sequences that include target scale variations, Struck$_{\text{UAV}}$ outperforms all other trackers as shown in Fig. 9. Although MEEM is the top-performer in overall OPE and SRE, Struck$_{\text{UAV}}$ outperforms it by 30% and 36% in OPE and SRE respectively, on this portion of the dataset. This performance gap is attributed to Struck$_{\text{UAV}}$'s ability to adapt to the scale of the target over time (refer to Section II-C). Interestingly, MEEM under performs other trackers also in this case, namely TLD and CXT, which are not competitive overall.

Because of its runtime, its overall tracking performance, and its robustness to scale variations, we select Struck$_{\text{UAV}}$ to be the best candidate to mount in our UAV system. It is noteworthy to mention that the MEEM tracker is not only a tracker, but a framework for tracking, where its base components (called experts) can be changed for improved accuracy and/or robustness. For future work, we

plan to integrate Struck$_{\text{UAV}}$ as the base expert in the MEEM framework to further improve tracking performance, while keeping runtime suitably low.

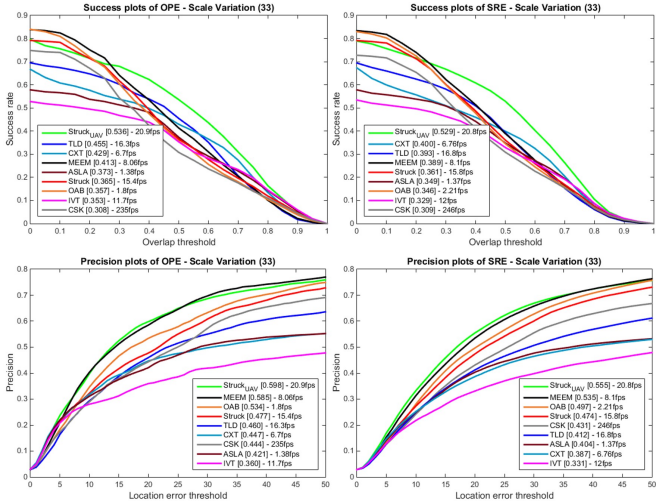

Fig. 9: (*top*) Success plots for OPE and SRE over sequences of our dataset that include scale variation (33 sequences in total). (*bottom*) Precision plots for OPE and SRE over the aforementioned sequences

B. Online Experiments

Proof of Concept. In order to qualitatively evaluate our implementation of the UAV tracking system, we conducted several online experiments. Figure 10 shows our setup. A wide open grass field (ca. 50 x 50m) was selected with plenty of room for the UAVs to maneuver at low altitude (10-15 m) and for our tracked human targets to walk and run. The first UAV was flown manually to the center of the field and placed in GPS hold mode. The human target then was asked to go the center of the field until reaching the FOV of the UAV. From the ground station, we requested a frame from the UAV, in which we annotate a bounding box around the target to be used for tracker initialization on the UAV. After initialization, the Struck$_{\text{UAV}}$ module on the onboard computer commenced tracking and the UAV becomes fully autonomous at that point. In all our trials, Struck$_{\text{UAV}}$ persistently tracked the target and periodically transmitted to the ground station the tracked bounding box at each time stamp. To simulate a natural scenario that goes beyond lab-controlled experiments, the target human was asked to move in different directions, speeds, and pose. The UAV running Struck$_{\text{UAV}}$ was able to track the moving target reliably throughout the flight despite all the variations that occurred. Diagnostics on the onboard computer showed that only 80% of the processor was needed to perform all the computations for our tracking system.

Robustness to Rapid/Erratic Motion. In a second experiment, the target was asked to make abrupt turns in multiple directions. When the subject moved in this manner, the UAV was able to respond efficiently and maintain tracking but with noticeable latency. However, if the target moved abruptly towards the copter and sought to run under it, the UAV often lost the target, since it did not respond fast enough to the change in direction until the target crossed

Fig. 10: UAV system.

the center of the frame in the opposite direction. The latency and poor response in this experiment is primarily related to the dampening employed on the UAV to provide smooth movement and avoid aggressive maneuvers. Struck$_{\text{UAV}}$ itself maintained the track until the target left the frame. Target re-initialization was successful in some trials and tracking was resumed, but it did fail in others.

Robustness to Occlusion. In a third experiment, we tested the effect of occlusion by allowing a second person interact with the target. This person walked with, in front, and behind the target, leading to varying degrees of partial occlusion to be addressed by Struck$_{\text{UAV}}$. This type of occlusion was handled well by the tracker, which kept a tight bounding box around the target even though the second person was very close. Tracker drift did not occur in this case.

Camera Handover. In the final experiment, we tested the camera handover module (refer to Section II-E), with the target moving in a natural manner similar to the first and third experiments. In this case, a second UAV is manually flown to the corner of the field and placed in GPS hold. Instead of exhausting the entire battery and then sending the handover request, we implemented a simple button on the ground station to trigger the handover function that in real-world scenarios would be triggered when the battery voltage monitor reached a low-level. This allowed us to test camera handover multiple times within the same flight. The handover was successful in this experiment. Even though the position and orientation passed to the second copter was limited by the accuracy of the GPS and compass, Struck$_{\text{UAV}}$ was still able to initialize the target and resume tracking even though the viewpoint of the new UAV was not the same as that of the first UAV. Videos of these experiments are included in the **supplementary material**.

IV. Conclusion

In summary, we provide extensive empirical evidence validating our proposed method and integrated system. We show that Struck$_{\text{UAV}}$ mounted on an off-the-shelf UAV is capable of persistently tracking a target and navigating accordingly to keep it centered in the visual field-of-view of the onboard camera. Within the current framework it is straight forward to further improve performance by implementing new trackers such as MEEM, determining velocity and heading of the object being tracked to reduce the search space, and by establishing communication between the UAVs to coordinate themselves irrespective of a ground station.

REFERENCES

[1] "Vot2015. http://www.votchallenge.net/vot2015/dataset.html."

[2] Y. Wu, J. Lim, and M.-H. Yang, "Online Object Tracking: A Benchmark," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2013, pp. 2411–2418.

[3] K. Kaaniche, B. Champion, C. Pegard, and P. Vasseur, "A Vision Algorithm for Dynamic Detection of Moving Vehicles with a UAV," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, April 2005, pp. 1878–1883.

[4] S. Ali and M. Shah, "Cocoa - tracking in aerial imagery," in *Proc. Int. Conf. on Computer Vision*, 2005.

[5] A. Qadir, J. Neubert, W. Semke, and R. Schultz, ser. Infotech@Aerospace Conferences. American Institute of Aeronautics and Astronautics, Mar 2011, ch. On-Board Visual Tracking With Unmanned Aircraft System (UAS), 0.

[6] D. Holz, M. Nieuwenhuisen, D. Droeschel, M. Schreiber, and S. Behnke, "Towards multimodal omnidirectional obstacle detection for autonomous unmanned aerial vehicles," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-1/W2, pp. 201–206, 2013.

[7] I. Cohen and G. Medioni, "Detecting and tracking moving objects for video surveillance," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, 1999, p. 325 Vol. 2.

[8] P. Campoy, J. Correa, I. Mondragn, C. Martnez, M. Olivares, L. Mejas, and J. Artieda, "Computer vision onboard uavs for civilian tasks," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 105–135, 2009.

[9] Q. Yu and G. Medioni, "Motion pattern interpretation and detection for tracking moving vehicles in airborne video," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2009, pp. 2671–2678.

[10] A. Nussberger, H. Grabner, and L. Van Gool, "Aerial object tracking from an airborne platform," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, May 2014, pp. 1284–1293.

[11] ——, "Robust aerial object tracking in images with lens flare," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 6380–6387.

[12] I. F. Mondragon, P. Campoy, J. F. Correa, and L. Mejias, "Visual Model Feature Tracking For UAV Control," in *2007 IEEE International Symposium on Intelligent Signal Processing*. IEEE, Oct 2007, pp. 1–6.

[13] I. F. Mondragon, P. Campoy, M. A. Olivares-Mendez, and C. Martinez, "3D object following based on visual information for Unmanned Aerial Vehicles," in *IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, 2011 IEEE*. IEEE, Oct. 2011, pp. 1–7.

[14] F. Lin, B. M. Chen, K. Y. Lum, and T. H. Lee, "A robust vision system on an unmanned helicopter for ground target seeking and following," in *2010 8th World Congress on Intelligent Control and Automation*. IEEE, July 2010, pp. 276–281.

[15] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying UAV," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, September 2011, pp. 4929–4934.

[16] A. Kendall, N. Salvapantula, and K. Stol, "On-board object tracking control of a quadcopter with monocular vision," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, May 2014, pp. 404–411.

[17] P. Doherty and P. Rudol, "A uav search and rescue scenario with human body detection and geolocalization," in *AI 2007: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, M. Orgun and J. Thornton, Eds. Springer Berlin Heidelberg, 2007, vol. 4830, pp. 1–13.

[18] A. Gaszczak, T. P. Breckon, and J. Han, "Real-time people and vehicle detection from uav imagery," in *IST/SPIE Electronic Imaging*, J. Röning, D. P. Casasent, and E. L. Hall, Eds., vol. 7878. International Society for Optics and Photonics, January 2011, pp. 78 780B–78 780B–13.

[19] J. Portmann, S. Lynen, M. Chli, and R. Siegwart, "People detection and tracking from aerial thermal views," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 1794–1800.

[20] J. Prokaj and G. Medioni, "Persistent tracking for wide area aerial surveillance," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1186–1193.

[21] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S.-C. Zhu, "Joint inference of groups, events and human roles in aerial videos," in *CVPR*, 2015.

[22] Y. Iwashita, M. Ryoo, T. J. Fuchs, and C. Padgett, "Recognizing humans in motion: Trajectory-based aerial video analysis," *BMVC2013*, pp. 127–1, 2013.

[23] J. Pestana, J. Sanchez-Lopez, P. Campoy, and S. Saripalli, "Vision based gps-denied object tracking and following for unmanned aerial vehicles," in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, Oct 2013, pp. 1–6.

[24] C. Fu, A. Carrio, M. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, "Robust real-time vision-based aircraft tracking from unmanned aerial vehicles," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 5441–5446.

[25] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection." *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, Dec 2011.

[26] B. Babenko, M.-H. Yang, and S. Belongie, "Visual Tracking with Online Multiple Instance Learning." *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, Dec 2010.

[27] R. Collins, X. Zhou, and S. K. Teh, "An open source tracking testbed and evaluation web site," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), January 2005*, January 2005.

[28] "Spevi. http://www.eecs.qmul.ac.uk/ andrea/spevi.html."

[29] "Caviar. http://groups.inf.ed.ac.uk/vision/caviar/."

[30] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2006, pp. 6.1–6.10, doi:10.5244/C.20.6.

[31] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[32] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *2011 International Conference on Computer Vision*. IEEE, Nov 2011, pp. 263–270.

[33] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Springer Berlin Heidelberg, 2012, vol. 7575, pp. 702–715.

[34] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 1822–1829.

[35] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 1177–1184.

[36] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 1838–1845.

[37] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: robust tracking via multiple experts using entropy minimization," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2014.

[38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[39] P. Krhenbhl and V. Koltun, "Geodesic object proposals," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, vol. 8693, pp. 725–739.

[40] T. V. Nguyen, "Salient object detection via objectness proposals," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[41] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What makes for effective detection proposals?" *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[42] "Chris veness. http://www.movable-type.co.uk/scripts/latlong.html."